

# IntelligentPadを用いた核反応データベース利用環境の試作

## A newly designed platform for the NRDF with IntelligentPad

札幌学院大学社会情報学部

千葉 正喜

Masaki CHIBA

Department of Social Information

Sapporo-Gakuin University

The NRDF database system is re-designed and developed applying IntelligentPad architecture. For the storage and management of NRDF data entry, one of the general corporate DBMS of a popular relational type is used. In the new system, every component of the user interface is represented by reactive media objects called Pad. The exploitation of the IntelligentPad architecture in developing the system enhances the usability and reusability of the tools developed. Several pads specially designed for the NRDF database system are shown together with the examples of how to use them.

### 1. はじめに

NRDF(Nuclear Reaction Data File)による荷電粒子核反応データの収集と蓄積が、1980年の開始以来継続して行われていて、すでに60MBのデータ量を蓄積している [1]。また、NRDFのデータをEXFORフォーマットに変換するシステムが開発され [2]、核反応データの国際交換にも参加して他の核データセンター等でコンパイルされたEXFORフォーマットの荷電粒子核反応データが得られるようになっている。

このように利用可能なデータ量が増大するにつれて、典型的な検索利用に加えて蓄積されたデータ全体をある側面で捉える、あるいはデータ相互間の関連に対する質問要求が増大してきている。検索結果として取り出したデータをその後の処理につなげることも必要とされてる。例えば、幾つかの数値テーブルデータの比較、部分的な観測データのある範囲での併合、または理論カーブとの比較などである。このように蓄積されたデータを前提に、新たに元素生成過程の研究あるいはデータの評価の活動も行われようとしている。収集されたすべての荷電粒子核反応データを利用するには、当然NRDFとEXFORにそれぞれ蓄積されたデータを同じシステムで同様に使えるようにする必要性もでてきている。

NRDFの「データ格納と検索サービス」システムは1975年に第一版が、その後第2版が開発され現在に至っている。このシステムは2層の索引ファイルとセット構造のデータモデルの基づくユニークなシステムである [3, 4]。データ収集等の周辺機能に追加や更新が行われたが、検索利用がメインフレーム上で実行されるというシステムの主要な部分には変更は無い。

このシステムは、実験数値データの収集蓄積という特徴があるが、大量に蓄積したデータの集合の中からユーザの特定の質問に応答して異なる事実を効率よく検索するという典型的な検索システムの見方にたっている。すなわち、記述的情報に基づく索引から条件を満足するデータテーブルを検索して、データテーブルとその関連記述情報の全体を出力するというものである。

それゆえ、このNRDFに対して、マルチメディア技術、オブジェクトオリエンテッド・モデリングとその開発技術、また高性能ワークステーション、インターネットの普及など今日のハードウェアとソフトウェア技術を取り入れた新しい概念に基づくユーザサービスシステムを設計する必要に迫られている。

そこでNRDFの新しいユーザサービスシステムの設計にあたり、IntelligentPadの技術を採用することにした。IntelligentPadは北海道大学で開発されている新しいオブジェクトオリエンテッドメディアシステムである [5, 6, 7, 8]。IntelligentPadは、モデリング、マルチメディアドキュメント、システム機能やアプリケーションプログラムの表現と統合、または管理に対して一様な枠組みを提供し

ている。IntelligentPadでは、これらの知的な資源がすべてパッドといわれる紙のイメージで表現され、これらのパッドは別のパッド上に貼られたり、剥がされたりして、それぞれ関連した機能を発現させる。さらに、ネットワーク環境のなかでミームメディアとしての研究が進められている。

このIntelligentPadアーキテクチャーを用いた新たなNRDFシステムでは、上に述べた課題の解決を図ること、またはそれにつながるようになることを期待している。

- (1) データの管理に一般的なデータベース管理システムを利用する
- (2) 典型的な検索機能に加えて、データベースに対するビューイング、フォーカシング、フィルターリングの繰り返し適用して課題意識の現実化を支援する
- (3) 検索結果とそれ以降の取り扱いに繋げる

以下では、このIntelligentPadのアーキテクチャに基づいて設計したNRDFデータベースのデータ管理とユーザインターフェースについて述べる。

## 2. IntelligentPadとデータベースproxy

まず、新しく設計するNRDFシステムは、IntelligentPadのアーキテクチャーに基づいているので、このIntelligentPad自身の概略を記述することにする。

### 2. 1 IntelligentPadの概要

IntelligentPadシステムでは、計算機上で取り扱うすべてのオブジェクトを紙としてのイメージを持ったメディアで統一的に扱う。この紙のイメージのメディアはパッド(Pad)と呼ばれる。IntelligentPadシステムでは、数値、文字列、グラフなどのさまざまな種類のデータがパッドで表現される。これらのデータだけでなく、アプリケーションプログラムもまたパッドで表現される。したがって、計算機上に表現されるあらゆるオブジェクトがパッドという表現形態で統一的に表現される。

ユーザはこのように表現されたパッドに対してマウスを操作して機能を発揮させること、または個々の機能を持つパッドを張り合わせて、複雑な機能を合成する

ことができる。紙のイメージ一枚のパッドはプリミティブパッドと呼ばれる。幾つかのプリミティブパッドが張り合わされてある機能を発揮するようになっているパッドは合成パッドと呼ばれる。パッドはその大きさや表示色を選択すること、また張り合わせるときの位置と大きさも自由にレイアウトできる。

## 2. 2 IntelligentPadの構造

IntelligentPadシステムはプリミティブパッドの集合であるが、各プリミティブパッドの構造はModel,View,Controllerの三つ組みからなるMVCプログラミングアーキテクチャに基づいている。Modelはパッドの内部状態を定義している。Viewはパッドの位置、大きさ、色など、パッドのディスプレイ画面上の表示形態を定義している。Controllerはユーザやシステムの操作に対するパッドの反応を定義するが、これはIntelligentPadシステムとして標準化されている。したがって、新しい機能を持つプリミティブパッドを定義しようとするときは、Model部とController部をプログラミングすることになる。

## 2. 3 IntelligentPadの機能合成機構

IntelligentPadシステムでは任意のパッドを貼り合わせることができる。幾つかの適当な機能のプリミティブパッドを貼りあわせて、新しい機能の合成パッドとすることができる。この貼り合わせ関係は、ちょうど電気のプラグをソケットに結合するように、標準化されている。IntelligentPadの各パッドには幾つかのスロットがあり、パッドはこのスロットにデータを保持している。このスロットの内容がパッドの内部状態を定義していることになる。貼り合わせたパッド間には親子関係が定義される。この貼り合わせで、上に置かれたパッドは子パッドと、子パッドの下に置かれたパッドは親パッドと呼ばれる。子パッドの一つのスロットのみが親パッドに幾つかあるスロットの一つに結合される。キボードからデータが入力されるなどのイベントが子パッドで発生して、親パッドに接続されているスロットの値が変更になると、子パッドは親パッドに対してsetメッセージを送る。このsetメッセージによって、子パッドのスロットの値は親パッドのスロットに送られる。逆に、親パッ

ドが子パッドにデータを送る必要があるときは、次の手順がとられる。親パッドの状態が変化したときは、子パッドに対してupdateメッセージを送る。子パッドはこのメッセージに対して反応して、gimmeメッセージを親パッドに送って親パッドのスロットの値を子パッドのスロットに読み出す。

このパッドの接続関係、すなわち子パッドのスロットを親パッドのどのスロットに接続するか、いずれのメッセージを通すかの指定は、コネクションシートを用いて各子パッドについて定義される。

## 2. 4 IntelligentPadの複製機能

IntelligentPadシステムでは、各パッドは自由に複製をつくることができる。複製の作り方に、共有コピーと非共有コピーの2種類がある。非共有コピーでは、複製されたパッドは元のパッドとまったく同じものがつくられ、両者は独立に動作する。一方、共有コピーで複製された場合には、複製されたパッドと元のパッドで台紙となっているパッドのモデル部が共有される。したがって、この場合は二つのパッドで状態を共有することになる。

## 2. 5 データベースproxyPadとデータベース管理

データベースや制御器など、IntelligentPadシステムの外部オブジェクトにアクセスするためのインターフェース機能を持っているパッドは総称してproxyPadと呼ばれる。proxyPadには、それぞれの外部オブジェクトに対するアクセスに必要な変数やアクセスの結果得られた値などをパッドのスロットとして保持するように作られる。proxyPadはパッド一般の機能も備えているので、任意のパッドと貼り合わせて合成パッドを作り、外部オブジェクトを含む必要な機能を実現することができる。

Smalltalk-80版IntelligentPadには、UniSQLに対するインターフェースプログラムが組み込まれている [9]。すなわち、Smalltalk環境がUniSQLのヴァーチャルマシンになっている。UniSQLは、リレーショナルモデルとオブジェクト指向データモデルを統合するデータベース管理システムである [10]。UniSQLでは、基本的なデータ型に加えて、集合や配列のような値のコレクションなども定義、クラス階層

とインヘリタンスを含むオブジェクト指向データ定義もできるなど、リレーショナルモデルからの拡張機能が含まれている。データベースのアクセスでは、リレーショナルDBMSの標準言語であるSQL文が使える。

Smalltalk-80からUniSQLをアクセスするために組み込まれたプリミティブ関数としては、

- (1) データベースファイル名に定義されるクラス名を返す

(UNISQL getClass)

- (2) クラス名に定義される属性名と型を返す

(UNISQL getAttribute: aClass)

- (3) SQL文に対する結果を返す

(UNISQL execSQL: sqlContext | UNISQL getForm: sqlContext)

機能が用意されている。

### 3. データベース管理システムの利用

その後のソフトウェア技術の進歩の一つは、データベース管理システムが一般的に利用できるようになったことである。現在広く使われているデータベース管理システムには、リレーショナルモデルに基づくデータベース管理システムがある。オブジェクト指向やデータのセットを扱うように拡張されているものもある。

現行のNRDFのデータベース管理の部分は、独自の開発したものであるが、事実上その保守は不可能になっている。このデータベースの管理の部分をこれらデータベース管理システムにまかせることにして、検索機能もデータベース管理システムのサービスを受けるようにする必要がある。この場合、データベースで管理するデータが時間的に不変なスキーマで捉えられること、実装するDBMSを用いてそのスキーマを確定すること、およびデータ入力プログラムの開発が課題である。

#### 3. 1 値属性対集合モデル

ここでの問題は、NRDFのデータの記述をいかにデータベース管理システムがサポートしているデータ構造に変換できるかである。NRDFでは、格納するデータ

の論理的単位はデータセットと呼ばれる。これは、数値データテーブルとその物理的意味内容を特徴づける情報または書誌的内容を区別する情報などを関連させたものである。これらの情報は、データ言語と呼ばれる規則に従って記述されるが、その基本は、〈属性〉=〈値〉の形式で表現される。発展の過程にある研究分野では〈属性〉の集合は時間的に固定できないとして、NRDFは、属性が取りうる〈値〉だけでなく、〈属性〉自身を任意に追加してよいことになっている。また、ひとつのデータセットの記述で使われる属性の集合は、その時点で定義されている属性の集合の一部であり、すべての属性の値が定義されるわけでもない。

リレーショナルスキーマでは、属性は時間的に変化しない。したがって、NRDFデータの記述で使われる属性をそのままリレーショナルスキーマの属性とすることができない。

この問題を解決するアイディアは、データオブジェクトをその要素データの集合として捉えることである。すなわち、ひとつの要素データは〈オブジェクト名、オブジェクト特性、特性値〉の三つ組みで表すと、一つのオブジェクトはオブジェクト名を共通に持つ要素データの集合で表すことができる。この三つ組みは時間的に不変なので、この三つ組みはリレーショナルスキーマとすることができる。そこで、データセットをオブジェクトとして、〈属性〉をオブジェクト特性に、〈値〉を特性値の対応させれば、NRDFデータは時間的に不変なスキーマで捉えることができる。したがって、このスキーマで記述されたデータはリレーショナルDBMSに格納することができる。ここで、オブジェクトをその要素データの集合で記述してリレーショナルDBMSで管理しようとするデータモデルを、値属性対集合モデル(VAP=ValueAttributePairSetModel)と呼ぶことにしよう。

### 3. 2 UniSQLデータ構造の定義

NRDFとEXFORには、それぞれ数値データを収集蓄積するフォーマットとそれぞれのデータ収集に使用されるコードの辞書のフォーマットがある。NRDFとEXFORのデータベースを統合して利用しようとする場合は、これらのデータをすべてUniSQLで管理する必要がある。辞書データについては、時間的に不変な属性の

集合で整理されている。NRDFの数値データについては、前節で考察したようにこれを値属性対集合モデルで捉える必要がある。EXFORデータは時間的に固定属性集合でデータが収集されているように見えるが、こちらはやはりNRDFと同様に値属性対集合モデルで考えるべきであろう。UniSQLで管理するNRDF、EXFOR、NRDF辞書、EXFOR辞書のデータ定義を付録1に於ける。定義にあたっては、UniSQLの持つクラス階層、コレクション属性の機能を積極的に用いている。

### 3. 4 データ入力プログラム

NRDFとEXFORのそれぞれの数値データと辞書データに対して、ファイルからデータベースのクラスに入力するLoadプログラムを開発している。これら4本のプログラムはSmalltalk [12] でプログラミングされている。それぞれModelのサブクラスに定義されている。

LoadNRDF	NRDFのデータストリームを読んでNrdfValueIndexとNrdfSectionのクラスのロードする
LoadNrdfDic	NRDFのデータ辞書をファイルから読んで、それぞれのクラスに格納する
LoadEXFOR	EXFOR Transテーブルファイルを読んで、ExforValueIndexとExforSectionクラスのロードする
LoadExforDic	辞書Transファイルを読んで、それぞれの辞書のクラスに格納する

## 4. パッド機能の設計

### 4. 1. 設計の概要

NRDFの数値データは、データベース管理システムUniSQLのクラスとして格納している。これらデータベースに対するアクセスはデータベースproxyPadを介して行う。したがって、データベースアクセスを伴う機能は、このパッドを台紙にしていくつかのプリミティブパッドを貼りあわせて実現していくことになる。検索結果

は、データセット識別子パッドまたはデータセットパッドとして取り出し表示される。データセット識別子パッドはデータセット識別子を、またデータセットパッドはデータセットをそれぞれオブジェクトとして保持するパッドである。検索結果としてのこれらのパッドは、その後さらにビューイングその他の機能のパッドと貼りあわされて必要な処理の対象にされることになる。以下では、実現した機能、その機構を実現するパッドの構成、および新しく開発したパッドについてはその機能の詳細の説明をつけ加える。まず、機能実現に共通に使われる幾つかのパッドや部品のパッドを挙げておく。

## 4. 2. 共通部品パッド

### (1) データベースプロキシパッド

パッド：PadView/NucDBViewerPadModel

機能：データベースを検索して機能を実現する合成パッドの台紙になる。

NucDBViewerPadModelは、MNFormViewerPad(中川光紀：シンセティックメディアによるフォームフローシステムの構築)を複写し、このデータベースアクセス専用の一部を調整したものである。このパッドの各スロットには次のようにパッドを結合する。

スロット：

- |         |   |
|---------|---|
| query   | NrdfValueIndexクラスまたはNrdfSectionクラスにたいするSelect文を与える。 |
| dbFile  | "NRDU"に設定してある。                                      |
| result  | 検索結果を保持している。データセット識別子生成配置パッドを結合する。                  |
| account | 検索結果の件数を保持している。数値表示パッドを結合すると、検索されたデータの件数を表示できる。     |
| search  | ボタンパッドを結合して、検索を開始させる。                               |

### (2) データ入出力パッド

パッド：LineEditPadView|PromptPadView|NumberPadView/PadModel

キーボードからスロットへのデータ入力あるいはスロットのデータ表示する場合に利用する部品パッドである。ビューを次のように使い分ける。

LineEditPadView： 比較的長い文字列で、カーソルを用いてその一部を修正することがあるような場合

PromptPadView： 比較的短い文字列の場合

NumberPadView： 数の場合

### (3) ボタンパッド

パッド：ButtonPadView/ButtonInputM

マウスを用いて、スロットに処理開始等のトリガーの信号を送るためのパッドである。データベースの検索処理を開始する場合などで使われる。

### (4) コンテナパッド

パッド：ContainerPadView/PadModel

このパッドはある複合パッドのコンテキストでこのパッドが覆っている下のパッドを積み込んで、別のコンテキストに運ぶ。データセット識別子パッドを一つの検索表示のコンテキストから他の検索表示のコンテキストへ幾つかまとめて移動するとき用いる。すなわち、フィルタリングの際に利用する。

## 4. 3. データセット識別子パッドとデータセットパッド

NRDFデータの論理的な単位はデータセットで、このデータセットはデータセット識別子で一意的に識別される。データセット識別子パッドはこのデータセット識別子を保持するオブジェクトである。データセットパッドは、一つのデータセット全体の記述を保持するオブジェクトである。データセット識別子パッドは、NrdfValueIndexクラスに対する検索の過程で主要な役割を果たす。データセットパッドは、ユーザの目的とするデータそのもので、その後の加工の対象となる。

### (1) データセット識別子パッド

パッド：NucDatasetIdPadView/NucDatasetIdPadModel

機能：このパッドは、データセット識別子生成配置パッドにより生成され、データセット識別子のリストを保持する。パッド上には、"DatasetIdPad"の表示およびデータセット識別子リストの1番目の要素、x属性値、y属性値の表示がある。しかし、データセット識別子生成配置パッド上に配置されたときは、10@10の大きさにしてあるので、これらの表示は見えない。それで、selectボタンが押されたとき、または他に移動されたときは90@90の大きさに拡大してパッド上に表示したこれらの情報を見えるようにしている。このパッドは、SectionクラスSelect文作成パッドに貼られ、データセットパッド生成に使われることを想定している。

スロット：

value:	データセット識別子リスト
xValue	x軸に対応づけられている属性の値
yValue	y軸に対応づけられている属性の値
xCoordinate	x軸上の位置座標
yCoordinate	y軸上の位置座標
rOrigin	selectボタンが押されたときに拡大して表示するとき、このパッドを移動するデータセット識別子生成配置パッド上の位置

## (2) データセット・パッド

パッド：NucDatasetPadView/NucDatasetPadModel

機能：このパッドは、一つの数値データテーブルとパラメータ情報、およびこのデータテーブルの理解に必要な実験条件と書誌情報のすべてをそれぞれのスロットに保持する。このパッドは、データセット生成パッドにより生成される。

スロット：

value	データセット識別子を保持する。これはd#とデータセット番号からなる。 <データセット識別子>：=<libID><d #>、<データセット番号>
-------	--

	⟨libID⟩ := <英字>
	⟨d#⟩ := <整数>
	⟨データセット番号⟩ := <整数>
bibSection	BIBセクションの記述情報
expSection	EXPセクションの記述情報
dataSection	DATAセクションの記述情報
dataTable	数値データテーブル

#### 4. 4. NrdfValueIndexクラスの検索

このクラスを検索する目的は、大きく二つあるだろう。一つは、条件を満足するデータセットを見つけだすこと、他の一つは、蓄積されているデータ全体またはそのサブセットをいろいろな側面または切り口でみること、または複数の側面の相関をみること、そしてそこから何らかの情報または知識を得ることである。この際、ビューイング、フォカシング、フィルタリングの過程を容易に繰り返し適用できるようにすることである。

そこで、このクラスのデータをアクセスする機能として、次の4機能を用意する。

- (1) 現時点で保持するデータに定義されている属性の集合を取り出すこと、
- (2) 特定の属性を指定して、その属性が取っている値の集合を取り出すこと、
- (3) 指定の属性が特定の値を持つデータセットの集合について、一つまたは二つの属性について、値の分布を表示すること、
- (4) 任意に選択したデータセットの集合について、(3)と同様にその分布を表示すること。

##### 4. 4. 1 属性、属性値の集合とメニュー選択パッド

属性またはある属性に対する値の集合があつて、そのメニューから一つに属性あるいは値を選択する機能を持つパッド（メニュー選択パッド）を考える。このパッドは、属性または値を一覧すること、およびデータ入力またはその補助手段として

も使える。

その時点において値を定義している属性の集合およびある属性に定義されている値の集合は、NrdfValueIndexクラスに対して問い合わせれば求めることができる。それで、検索結果に対して属性の集合と値の集合のそれぞれについて、これを編集してメニュー選択パッドに渡す機能のパッド（属性メニュー設定パッド，属性値設定パッド）とメニュー選択パッドを用意すればよい。

#### (1) 属性メニュー設定パッド

パッド：NucAttribMenuSetPadView/PadModel

機能：DBViewerPadの結果スロットに接続される。このパッドにメニュー選択パッドが貼られると、resultの内容からattribに対応する値を取り出し、メニュー選択パッドのmenuListスロットに設定する。なお、この場合、DBViewerPadのqueryスロットには次のクエリが設定する。

```
select distinct attrib from NrdfValueIndex;
```

#### (2) 属性値メニュー設定パッド

パッド：NuctValueMenuSetPadView/PadModel

機能：DBViewerPadの結果スロットに接続される。このパッドにメニュー選択パッドが貼られると、resultの内容からtValueに対応する値を編集して、属性選択パッドのmenuListスロットに設定する。

DBViewerPadのqueryスロットには、次のような値を設定する。

```
select distinct tValue from NrdfValueIndex where attrib='ATH';
```

ここで、'ATH'のような属性はこのselect文のパラメータであるから、属性をパラメータにselect文を作成するパッド(PadView/NucValueIndexSelectVModel)をこのスロットに接続する。

#### (3) メニュー選択パッド

パッド：NucMenuSelectorPadView/NucMenuSelectorPadModel

機能：このパッドはSelectボタンに反応して、menuListスロットの値をポップアップメニュー表示して、選択された値をプライマリスロット(value)に設定する。

menuKindスロットの内容を自身に表示する。

スロット：

menuList	メニュー表示する値の集合を保持する
menuKind	メニューの値の種類を表す文字列
value	メニューから選択された値が設定される

#### 4. 4. 2 データセット識別子オブジェクトの検索と表示

属性、属性と値の対の幾つかの条件から、データセット識別子オブジェクトを検索する機能を実現する。検索結果のデータセット識別子をデータセット識別子パッドとして生成する。このデータセット識別子パッドを指定の一つまたは二つの属性を軸にとり平面または直線上にデータセット識別子パッドを配置する。ビューイング、フォーカシング、フィルタリングの過程を、データセット識別子オブジェクトを介して繰り返し実行できるように、このオブジェクトを一時的に保存するデータベースクラスAndSetWorkを使用する。

この機能は、DBViewerパッドを台紙にして、ValueIndexクラスSelect文作成パッド、検索結果配置パッド、および幾つかの入出力パッドと処理を開始させるボタンパッドを貼り合わせて実現される。

##### (1) データセット識別子選択Select文作成パッド

パッド：PadView/NucValueIndexSelectPadModel

機能：このパッドはpredicate 1 スロットとpredicate 2 スロットに入力された検索条件の記述からSQLのselect文を作成して、自身のvalueスロットに設定する。このパッドは、NucDBViewerPadのqueryスロットに結合する。

スロット：

predicate1	検索条件の指定
predicate2	検索条件の指定
value	このスロットにはpredicate1とpredicate2から作成したselect文が設定される。作成されたselect文を確認するには、このスロットに文字列が出力できるパッドを結合すればよい。

## 検索条件の指定

文字入力パッドを predicate1 スロットと predicate2 スロットに結合して、つぎの形式で検索条件を指定する。

属性名[**[**, 拡張記号の位置]**]**={ | like }値]

ここで、**[** と **]** は囲まれた部分は省略できる、**{ | }** は **|** で区切られた **{ }** 内のいずれかを選択することを示す。

この指定により、value スロットに作成される select 文は、次の形式のものである。

```
select dId,dIdNo,DataSetList,attrib,tValue from NrdfValueIndex where attrib = <属性名> [and evalSQ = <拡張記号の位置> [and tValue { = | like } <値>];
```

または

```
select x.dId,x.dIdNo,x.attrib,x.tValue,y.datasetList,y.attrib,y.tValue from (<predicate1から作られるselect指定>) as x(dId,dIdNo,datasetList,attrib,tValue),(<predicate2から作られるselect指定>) as y(dId,dIdNo,datasetList,attrib,tValue) where x.dId = y.dId and x.dIdNo = y.dIdNo and (x.datasetList superseteq y.datasetList);
```

### (2) データセット識別子 2 次元配置パッド

パッド：NucDataSetIdSpreadPadView/NucDataSetIdSpreadPadModel

機能：このパッドは、NucDBViewerPad の result スロットに結合する。result スロットからの検索結果の情報を value スロットで受け取り、これをデータセット識別子パッドにして、自身の上に貼る。検索条件に指定した属性に定義される値の順序にしたがって、データセット識別子パッドを配置する。例えば、元素は番号の上昇順、数値データはその大きさの順である。

スロット：

value	NrdfValueIndex クラスの検索結果
xSize	x 軸の属性値の異なる値の数
ySize	y 軸の属性値の異なる値の数

xAttrib	x 軸の属性名を保持する
yAttrib	y 軸の属性名を保持する
elmDic	元素の順序を定義する辞書を保持する

### (3) 検索集合一時保存パッド

パッド：NucAndSetWorkMakerPadView/NucAndSetWorkPadModel

機能：このパッド上に置かれたデータセット識別子はAndSetWorkクラスに挿入できる。setスロットに連結しているボタンを押すと、このパッド上に置かれているすべてのデータセット識別子パッドが保持しているデータセット識別子をAndSetWorkクラスの値とする。clearスロットに連結しているボタンが押されると、このパッド上に置かれているデータセット識別子パッドを消去する。

なお、データセット識別子パッドをそれぞれ個別に、またはコンテナパッドを用いると一括して、別のコンテキストから持ってくるができる。

スロット：

value	AndSetWorkクラスに書き出されるデータセット識別子の集合が設定される。
clear	このパッド上のデータセット識別子パッドの消去をはじめめる。
set	このパッド上のデータセット識別子パッドをAndSetWorkクラスに書き出す。

### (4) 検索集合一時保存パッド2

パッド：PadView/NucAndSetWorkPad2Model

機能：属性と値の対を定義しているデータセット識別子の集合をAndSetWorkクラスに一時保存する。storeスロットに連結されているボタンが押されたときに子の処理を開始する。clearスロットのボタンが押されたときは、保持しているデータセット識別子の集合をリセットする。

スロット：

attribValPair	属性と値の対の指定を保持する。このスロットには属性値対選択パッドが連結される。
clear	このパッド上のデータセット識別子パッドの消去をはじめめる。
set	このパッド上のデータセット識別子パッドをAndSetWorkクラスに書き出す。

#### (5) 属性値対選択パッド

パッド：NucAttribValuePairSelectPadView/

NucAttribValuePairSelectPadModel

機能：attribスロットに指定された属性に対して定義されている値の集合を検索して、マウスのセレクト操作に反応してこの値をメニューに表示し、選択された値を属性と対にしてattribValPairスロットに設定する。

スロット：

attrib 属性の指定を保持する。

attribValPair メニューの選択結果を'属性=値'の形式で保持する。

#### 4. 5. NrdfSection クラスの検索

このクラスに対する検索は、ほぼデータセットを構成する目的に限られよう。データセット識別子を入力して、このデータセット自身をパッドとして出力する。

その手順は、次のごとくである。データセット識別子パッドのデータセット識別子の情報からNrdfSectionクラスを検索するSQL文を生成する。次いで、検索演算を実行してひとつのデータセットを構成するセクションの集合を取り出して、この結果を編集してデータセットのすべて情報を保持しているデータセットパッドをジェネレートする。

この機能は、DBViewerパッドを台紙にして、(1) NrdfSectionクラスSQL文作成パッドをqueryスロットに、(2) データセット生成パッドをresultスロットに、ボタンパッド(GetDataset)をsearchスロットに結合して実現する。もし、検索結果の件数を表示する必要があるならばaccountスロットに表示パッドを結合すればよ

い。さらに、NrdfSectionクラスSQL文作成パッドの各スロットには、次のパッドを結合する。nextDataスロットにはボタンパッド(NextDataset)を結合する。dSetIdスロットに入出力パッドを結合すると、検索するデータセット識別子を入力または表示することができる。noOfDatasetスロットに出力パッドを結合すると、データセット識別子の個数を表示できる。valueスロットに出力パッドを結合するとselect文を表示できる。

この機能を動作させるには、まづデータセット識別子パッドをNrdfSectionクラスSQL文作成パッドにはる。そして、二つのボタン(GetDatasetとNextDataset)を交互の押して、データセットパッドを生成する。

#### (1) SectionクラスSelect文作成パッド

パッド：PadView/NucSectionSelectModel

機能：このパッドは、これにデータセット識別子パッドが貼られることが動作を開始の契機となり、データセット識別子のリストから一つずつデータセット識別子を取り出しデータセットを構成するために必要なNrdfSectionクラスに対するselect文を生成する。

データセット識別子パッドがこのパッドに貼られると、データセット識別子パッドのvalueスロットの内容、すなわちデータセット識別子のリストをdSetListスロットに設定して、その最初の要素をdSetIdスロットに移す。noOfDatasetスロットには、dSetIdListに残っているデータセット識別子の個数が設定される。dSetIdスロットに設定されたデータセット識別子の情報から次の形式のsql文を作成してvalueスロットに用意する。

```
select sectiono,sectionKind,sectionBody from NrdfSection where datasetList superseteq  
{<データセット番号>} and dIdNo = <d#> and dId = 'dlibId' ordered by  
sectionKind,sectionNo;
```

nextDataスロットに結合されているボタンがマウスで選択されると、データセット

識別子リストの次のデータセット識別子に対して、上に述べて動作を繰り返す。

スロット：

dSetIdList	データセット識別子リスト
dSetId	データセット識別子を保持し、これに値が設定されるとselect文を作成する。
noOfDataset value	dSetIdListスロットが保持しているデータセット識別子の個数 作成したNrdfSectionクラスにたいするselect文
nextData	ボタンパッドを接続する。ボタンが押されると、dSetIdListスロットから、次のデータセット識別子と取り出し、それぞれのスロットの値を更新する。

## (2) データセット生成パッド

パッド：NucDatasetGenPadView/NucDatasetGenpadModel

機能：NrdfSectionクラスに対する検索結果から、データセットパッドを生成する。

このパッドは、DBViewerパッドのresultスロットに結合され、検索結果であるひとつのデータセットを構成するセクションの集合を受け取る。この検索結果を編集して、BIBセクション、EXPセクション、DATA データセクションとデータテーブルごとに記述レコードをまとめ、データセットパッドのそれぞれのスロットに設定してパッドとしてを生成する。

スロット：

value	検索結果
-------	------

## 4. 6. データセットパッドに反応するパッド

個々のデータセットの記述内容を調べるため、または幾つかのデータセットを詳細に比較するには、一つまたは幾つかのデータセットパッドに反応して機能するパッドを用意すればよい。そのような例として、次の機能を持つようにパッドを用意している。

### (1) テキストビューワパッド

書誌情報，実験条件，実験パラメータ，データテーブルの一つまたはすべての記述情報を見る。

## (2) グラフパッド

このデータテーブルをデータ点のグラフとして表示する。

## (3) グラフ台紙パッド

複数枚のグラフを貼りあわせて，その様相を比較する。

## (1) テキストビューワパッド

パッド：NucTextViewerPadView/NucTextViewerPadModel

機能：データセットパッドが貼られると，その記述データを取りこみ，sectionKind スロットに指定のある場合は，指定のセクションのみを，指定のない場合はデータセット全体を行単位の文字列として表示する。表示すべき行数に応じて，パッドの大きさが変わる。

スロット：

value	データセット識別子を保持している。
sectionKind	入出力パッドを接続して，{bib exp data table  〈空白〉}のいずれかを指定する。
bibSection	このパッド内におけるグラフ用紙パッドの配置位置と 大きさ

## (2) グラフパッド

パッド：NucGraphPadView | NucGraphLinLinPadView |  
NucGraphLinLogPadView | NucGraphLogLinPadView |  
NucGraphLogLogPadView/NucGraphPadModel

機能：核反応実験データの典型的なものは，独立変数の値に対する観測値とそのエラーの大きさから構成されている。そのグラフ表示では，x座標に独立変数の値を取り，y座標に観測値とそのエラー値を取り観測値とそのエラー値が表示される。

座標のスケールはそれぞれリニアースケールの場合と対数スケールの場合がある。それで、グラフパッドとしては、次のスケールの組み合わせでグラフが描ける必要があり、Viewがそれぞれこのスケーリングに対応している。

スケーリング(x/y)	対応する View
リニア－／リニア－	NucGraphPadView   NucGraphLinLinPadView
リニア－／対数	NucGraphLinLogPadView
対数／リニア－	NucGraphLogLinPadView
対数／対数	NucGraphLogLogPadView

なお、観測値のパッド上の座標位置は、次のように計算している。

観測値(x,y)

観測値の最大値(xMax,yMax),最小値(xMin,yMin)

パッド上の表示座標(xPad,yPad)

パッド上の表示領域(xPadMin,yPadMin),(xPadMax,yPadMax)

とすると、

リニアースケールの場合：

$$(x-xMin)/(xMax-xMin) = (xpad-xPadMin)/(xpadMax-xPadMin)$$

よって、

$$xPad = (xPadMax-xPadMin)*(x-xMin)/(xMax-xMin) + xPadMin$$

ここで、

$$xPadMin = 0$$

$$xPadMax = (aPad extent) x$$

とすると、

$$xPad = (xPadMax)*(x-xMin)/(xMax-xMin)$$

$$(y-yMin)/(yMax-yMin) = (yPadMax-yPad)/(yPadMax-yPadMin)$$

よって,

$$yPad = yPadMax - (yPadMax-yPadMin)*(y-yMin)/(yMax-yMin)$$

ここで,

$$yPadMin = 0$$

$$yPadMax = (aPad \text{ extent}) y$$

とすると,

$$yPad = yPadMax \cdot (yPadMax*(y-yMin)/(yMax-yMin))$$

対数スケールの場合:

$$xPad = (xPadMax)*(log(x)-log(xMin))/(log(xMax)-log(xMin))$$

$$yPad = yPadMax \cdot (yPadMax*(log(y)-log(yMin))/(log(yMax)-log(yMin)))$$

データセットパッドがこのパッド上に貼られると、データテーブルのカラムヘッディング、単位、観測データ点の値をそれぞれスロットに格納し、観測点をパッド上にグラフ表示する。このパッド上のイベントは親パッドに送る。データ点は、もしエラーバーがあれば、これと共に表示する。表示されるデータ点は、それらの最小値最大値でスケーリングされている。グラフ台紙パッド上にある場合は、グラフ台紙パッド上のすべてのグラフパッドに表示されているすべてのデータの最小値最大値でスケーリングされる。

背景色は白またはグラフ台紙パッドの上の場合は透明となる。

スロット:

value	データセット識別子
dataValues:	数値データ点で、4つの値の組（独立変数の値、観測値、-エラー値、+エラー値）のコレクション
dataValueRange	このパッドが保持するデータ点の値の最小値と最大値の座標点

displayValueRange	グラフ台紙パッド上のすべてのデータ点の最小値と最大値の座標点で、このスロットの値はグラフ台紙パッドが設定する。
dataHeadings	データテーブルの各カラムに対するヘッディング
datanits	データテーブルの各カラムの数値の単位

### (3) グラフ台紙パッド

パッド：NucGraphBasePadView/NucGraphBasePadModel

機能：このパッド上には、一つまたは複数のグラフパッドを貼ることができる。貼られた各グラフパッドは背景色を透明にして、各グラフの座標のスケールを一致させて重ねて表示する。したがって、このパッドを用いるとグラフのかたちで、データセットを比較できる。

グラフ用紙パッドが貼られると、そのデータの最小値と最大値と取り込み、グラフ表示に使うべき座標の最小値最大値を再計算して、再計算された最小値最大値とグラフ表示領域の大きさを各グラフパッドに送る。

グラフパッドが剥がされたときは、そのパッドの最小値最大値データを削除し、表示座標を再計算して、各グラフパッドに送る。また、グラフ台紙パッドの大きさが変更されたときも、各グラフパッドに知らされる。

スロット：

dataRangeList	このパッドに貼られた各グラフパッドに表示されているグラフの座標の最大値最小値を保持している。
displayValueRange	このパッドに貼られた各グラフパッドがデータのグラフ表示に際してデータのスケールリングで使用すべき最小値最大値
graphArea	このパッド内におけるグラフ用紙パッドの配置位置と大きさ

## 5. IntelligentPadユーザインターフェース

設計開発したパッドおよびパッドシステムにある既存の部品のパッドを用いて、NRDFデータベースを利用するシステムのを具体的に構成し、以下に示す。

### 5. 1 ユーザインターフェースの概観

IntelligentPadシステムの一つのアプリケーションは、一つのウィンドウのルートパッド上に現れる。図1は、IntelligentPadを用いて開発したNRDFデータベース利用システム全体を示すディスプレイスナップショットである。この図には、NRDFデータベースシステムに利用されている汎用部品のパッドとこのシステムのために特別に開発されたパッドが表示されている。

図1には、パッドが3列に並べてある。左端の列は、このアプリケーションで使う部品とツールのパッドが並べてある。中央と右端の列には、このアプリケーションの機能を実行する幾つかの合成パッドを配置してある。

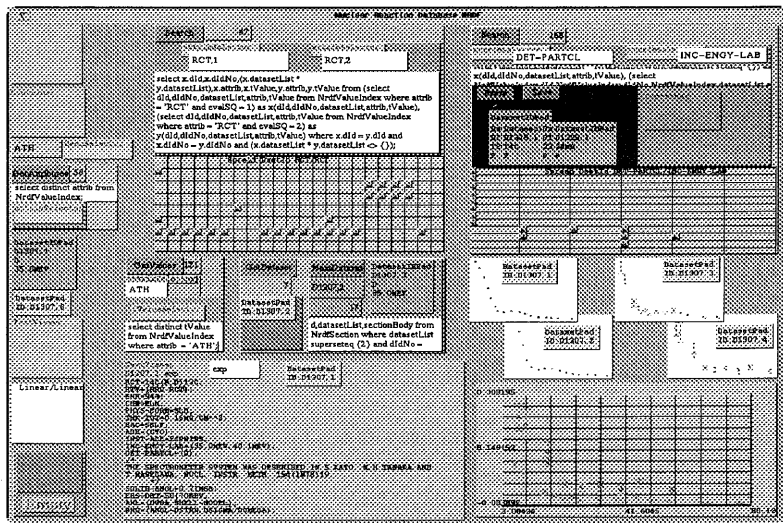


図1 WindowのRootPad上に配置されたNRDFシステムの全体

左端の列の上から順に "パッドメーカーパッド" , "コンテナーパッド" , "入出力パッド" と "メニュー選択パッド" , "属性メニュー設定パッド" , "データセット識別子パッド" , "データセットパッド" , "テキストビューワパッド" , それに任

意のパッドが消去できる "TrashPad" である。

データセット識別子パッドは、インデックスファイルを検索したときに作られ、データセットを代表する識別子を保持している。これに対して、データセットパッドは、一つのデータセットの記述データをすべて保持するパッドで、セクションファイルを検索して作られる。

中央と右の二つの列の上段の二つのパッドはデータベースのインデックスファイルを検索するための合成パッドである。中央の列の中段には二つのパッドがあるが、左側のパッドは、属性名に対する値の一覧をメニュー選択パッドに設定する合成パッドで、右側のパッドはセクションの集合から特定のデータセットを構成するための合成パッドである。中央の列の一番下には、データセットパッドを貼ったテキストビューパッドがある。右の列の中段には、データセットパッドを貼った4枚のグラフパッドがある。また、右の列の下段にあるのは、二つのグラフパッドを重ねたグラフ台紙パッドである。

以下で、これらの合成パッドの機能を説明する。

## 5. 2 データセットの属性面へのマッピングと多段階フィルターリング

図2のパッドはNRDFデータベースをアクセスして、レコードの分布状態を視覚的に表示する合成パッドである。このパッドの機能は一つのデータベースプロキシパッド [7] を台紙にして、4枚の入出力パッド、属性名のリストをメニューに設定した2枚のメニュー選択パッド、一つのボタンパッド、一つのSQLクエリ作成パッドと一つのレコード分布パッドから構成されている。

データベースを検索するにはまず、NRDFデータベースで情報を記述している属性の一つまたは二つを選択して、その属性名を指定する。指定できる属性の候補は、属性名のリストを設定したメニュー選択パッド(attribSelector)をマウスでセレクトすればポップアップメニューで調べ、その一つを選択できる。属性名の中から、図2では核反応式の情報を記述している属性名の "RCT" を指定している。この指定は、入力パッドに直接キーボードから入力する、あるいは入力パッドと同じスロットに結合してあるattribSelectorのポップアップメニューの助けを借りて行うことが

できる。核反応式の属性では、そこに書かれる情報の順序でその内容が区別されている。すなわち、記述される要素の順に、標的となる原子核（標的核という）、入射粒子、放出粒子、反応生成物として残っている原子核などを意味している。

”RCT,1”と”RCT,2”の指定は、それぞれ属性名RCTの1番目と2番目を指すことを表し、標的核と入射粒子を意味している。入力パッドでこのように指定されると、SQLクエリ作成パッドはこれらのパラメータ情報からSQLのselect文を作成する。このselect文はデータベースプロキシパッドのqueryスロットに渡される。もし、作成されたselect文を確認する必要がある場合は、図にあるように出力パッドをこのqueryスロットに接続すればよい。または、このパッドを入力器として用いれば、完全なSQLのselect文を入力することもできる。

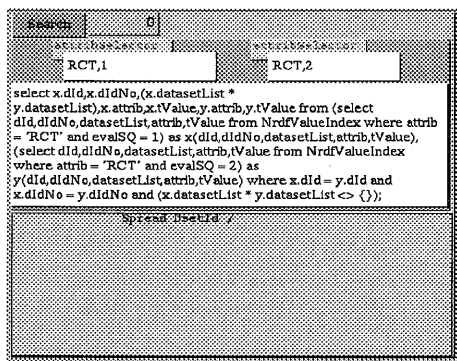


図2 索引ファイル検索の合成パッド  
二つの属性の指定とselect文が  
入出力パッドに示されている

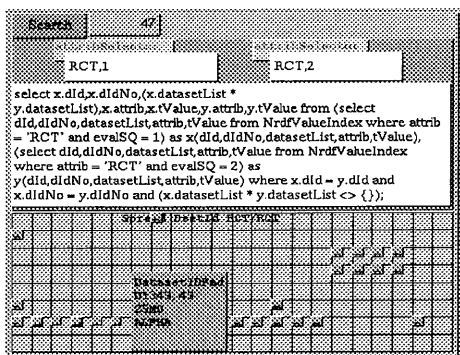


図3 データセット識別子パッドを配置した  
索引ファイル検索の合成パッド

searchボタンパッドをクリックすると、NRDFデータベースの検索を開始する。データベースの検索を終了すると、検索で見付かったレコードの件数が出力パッドに表示される。図3の例では、”47”となっている。これと同時に見付かったレコードはデータセット識別子パッドに造られてレコード分布パッド上に図3のように配置される。レコード分布パッドには、x軸とy軸がありそれぞれ入力条件の属性に対応して定義される値の順序をマッピングしている。したがって、この例の場合、x軸には標的核の種類が、y軸には入射粒子の種類が順番にマッピングされている。

レコード分布パッド上に配置されたデータセットパッドの縦の各列には同じ標的核のデータについてのデータセット識別子パッドが並び、また横の各行には同じ入射粒子についてのデータセット識別子パッドが並んでいる。データセット識別子パッドをマウスでセレクトすると、セレクトされたデータセット識別子パッドは十分な大きさに広がって、その座標位置に対応する属性の値を確認することができる(図3)。

いくつかの異なる属性に関してデータベースのレコードの分布を調べられる。図1の右端の列の上段にあるパッドは、"and条件集合作成パッド"が追加して貼られていることを除けば、機能的には今ちょうど上で述べたパッドと同じものである。この前の検索結果から適当に選択したデータの部分集合について、さらに別の属性でレコードの分布状況を調べられる。例えば、関心のある一部のデータについて、さらに検出した粒子と入射粒子のエネルギーに関してレコードがどのように分布しているか調べたいものとする。レコードを任意に取り出すには、コンテナパッドが使える。このパッドは、このパッドが覆っているパッドを自身に積み込む。すなわち、コンテナパッドをマウスを用いて移動し積み込みたいパッドを覆うように大きさを調整して、マウスのセレクトボタンを押す。そうするとポップアップメニューが現れるので、getオペレーションを選択すればよい(図4)。

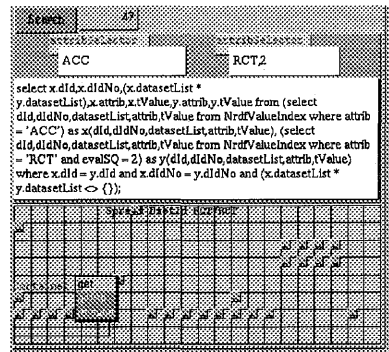


図4 データセット識別子パッドを積み込むコンテナパッド

データセット識別子パッドを積み込んだら、コンテナパッドをand条件集合作成パッドに移動し、コンテナパッドの上でマウスのセレクトボタンを押す。そうすると、再びポップアップメニューが現れるので、そこでreleaseオペレーションを指定する(図5は移動後示している)。積み込まれていたデータセット識別子パッドはコンテナパッドから解放され、and条件集合作成パッド上に貼られる。ここで、Saveボタンパッドをクリックして、データセット識別子パッドで運んできたデータセット識別子レコードをデータベースの作業ファイルに書き込む。



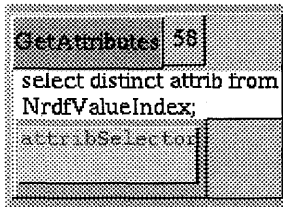


図7 MenuSelectorへの属性名リストの設定

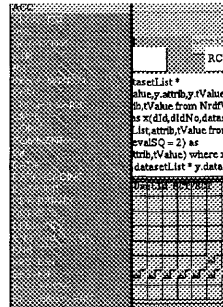


図8 attribSelectorのポップアップメニュー

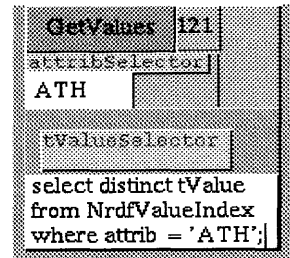


図9 MenuSelectorへの著者名リストの設定

属性名の集合をインデックスファイルからフィルターリングするのが”属性メニュー設定パッド”である。このパッドにメニュー選択パッドを貼る（図7）と属性名のリストをポップアップメニューで見ることとその値の一つを選択することができるようになる（図8）。

また、ある属性名に対する値の集合をインデックスファイルからフィルターリングするのが、図9のパッドである。この例では、著者（ATH）に対する値の集合を取り出す指定をしている。これも、メニュー選択パッドを貼れば、著者名のリストがポップアップメニューで見られるようになる。

#### 5. 4 データセット識別子からデータセット自身へのマッピング

データセット識別子パッドから、そのデータセットを記述している完全なデータをデータセットパッドとして取り出すには、図10に示す複合パッドを用いる。この図に示すように1枚の任意のデータセット識別子パッドをこのパッド上に貼る。データセット識別子パッドには、同じ文献番号でかつそれまでに指定された検索条件を満たすデータセット識別子レコードの集合が含まれている。一つのデータセット識別子パッドが、図10のようにこのパッドに貼られると、データセット識別子レコードの集合から1要素が取り出される。取り出された要素は、”D1307,1”のように表示される（図10）。このレコードの識別子コードは、パラメータとしてSQLのselect文に埋め込まれる。ここで、GetDatasetボタンパッドをクリックすると、セ

クシオンファイルからこのデータセット識別子に対するセクションレコードがフィルタリングされる。データセットパッドが作られると、図11にあるようにそれが現れる。データセット識別子レコードの集合の次の要素に対応するデータセットパッドを作るには、NextDatasetボタンパッドをクリックしたあと、上と同じ操作を繰り返せばよい。

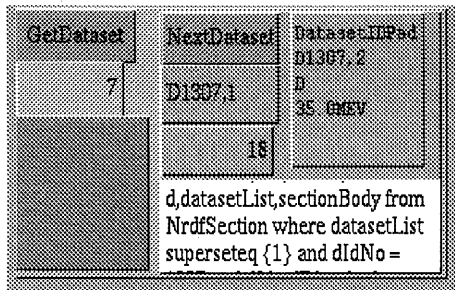


図10 データセット識別子パッドが貼られたデータセット作成パッド

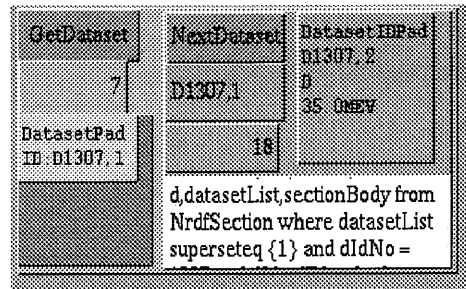


図11 データセットパッドを生成したデータセット生成パッド

## 5. 5 データセットパッドのビューイング

データセットパッドが得られたならば、目的に応じた適当なビューワパッドを用意すればデータセットの詳細を見ることができる。その例として、データセットの記述をテキストで表示する"テキストビューワパッド"、およびデータテーブルをグラフにして見るための2種類のパッドが利用できる。

テキストビューワパッド(図12)は、これにデータセットパッドを貼るとデータセットの記述データを表示する。このパッドの大きさはテキスト表示に十分な大きさに広がるが、マウスをクリックするたびに拡大と縮小を互いに繰り返す。テキストビューワパッドのsectionKindスロットには入力パッドが結合でき、表示テキストをフィルタリングする。"BIB"、"EXP"、"DATA"、または"TABLE"が指定されたときは、それぞれ書誌の情報、実験条件の情報、データテーブルのパラメータ情報、データテーブルのいずれかの記述テキストが表示され、このような指定のないときはデータセット全体が表示される。

データテーブルのグラフ表示の一つのパッドは"グラフパッド"で、もう一つ

は"グラフ台紙パッド"である。グラフパッドは、もしデータテーブルにエラーバーの欄があればそれをつけて、データテーブルを点グラフに描く。グラフ台紙パッドは、複数のデータテーブルを視覚的に比較するために使う。このグラフ

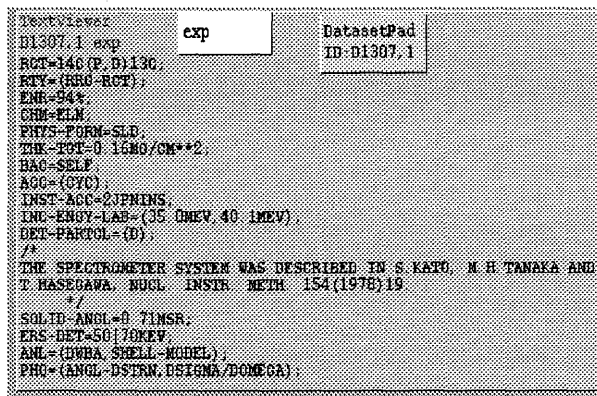


図12 データセットパッドが貼られたテキストビューワパッド

台紙パッドは、これに貼られる各グラフパッドの背景色を透明にし、このパッド上のすべてのグラフパッドの座標軸の尺度をx軸とy軸ごとに共通にする。図13には、その上部にそれぞれデータセットパッドを貼った4枚のグラフパッドを、下部には二つのデータセットD1301,3とD1301,4のそれぞれ対するの2枚のグラフパッドを重ねたグラフ台紙パッドを示している。

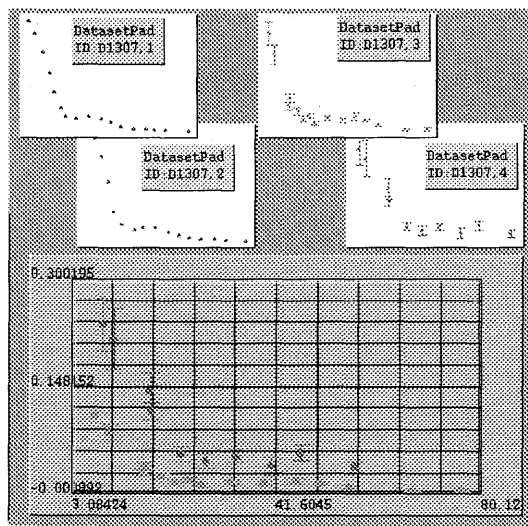


図13 データセットパッドが貼られた4枚のグラフパッドと2枚のグラフパッドが重ねられたグラフ台紙パッド

## 6. まとめ

IntelligentPadのアーキテクチャーを用いて新たに設計試作したNRDFデータベースの利用システムを示した。これは、今日の計算機のハードウェアとソフトウェアの環境で実行され、荷電粒子核反応データに対する利用要件を満たそうと試みたも

のである。

現に進歩発展している分野のデータであっても、値属性対セットモデルでそれを捉えれば、リレーショナルデータベース管理システムなど一般的な管理システムで管理できることを示した。研究環境におけるデータベースの利用は、仮説の設定、指向、結果の評価 (think-try-see)の繰り返しである。データベースの機能として、データベースをある側面のみでみる (viewing)、関心のあるデータに焦点を当てる (focusing)、特定のデータを取り出す (filtering)、視点を変更す (change of view) をスムーズに繋げられなければならないが、5節の例が示すようにIntelligentPadはパッドというオブジェクトがデータと手続きを統一的に扱ってそれを可能にしている。データに対する種々の解析ツールをパッドで用意すれば、使用例そのものが使用法のドキュメントとなっている。道具の再利用の観点から優れていると考えられる。

今までに用意できたパッドは、NRDFデータにたいするものである。データベース管理システムにロードしたデータもNRDFデータの一部に限られている。EXFORのデータを含むすべてのデータをデータベースファイルにローディングして、その機能、性能を確認することが今後の課題として残されている。このシステムが、データの発表、流通、利用を一貫してサポートすることを考慮することも必要である。

## 謝辞

1995年4月から1年間、著者は札幌学院大学および社会情報学部から研修員として北海道大学工学部に派遣され研究に専念することが許された。この仕事は、そのもとで行われたものである。

研修先の北海道大学工学部計算機情報通信工学講座の田中譲教授には、有益な議論と示唆をいただいた。中川光紀氏をはじめこの研究室の多くの方々には、システム開発で助力をいただいた。また、北海道大学理学部物理の原子核研究室の加藤幾芳、大西明を含む日本荷電粒子核反応データグループ管理運営委員会のメンバー、および札幌学院大学社会情報学部の原田融の各氏には、システムの利用条件とパッドの設計に際して議論いただいた。記して感謝の意を表したい。

## 参考文献

- [ 1 ] K. Kato, "Charged Particle Nuclear Reaction Database NRDF \_ Present status and its usage \_", 原子核研究, Vol.39, No.5, 1995.
- [ 2 ] M. Chiba, "A database translator of nuclear reaction data for international data exchange", Journal of Information Science, 12 (1986) pp153-165.
- [ 3 ] M. Togashi and H. Tanaka, "Scientific information sysytem for nuclear physics research", in Scientific Information System in Japan, H. Inose(ed.),1981, North-Holland.
- [ 4 ] M. Togashi and H. Tanaka, "An information system for charged particle nuclear reaction data", Journal of Information Science, Vol.4, No.5, 1982.
- [ 5 ] Y. Tanaka and T. Imataki, "IntelligentPad:A Hypermedia System Allowing Functional Composition of Active Media Objects through Direct Manipulations", Information Processiong 89. Proc. of the IFIP 11th World Computer Congress, 541-6, 1989.
- [ 6 ] Tanaka, Y. "IntelligentPad: a media-synthesizing tool kit for the management of active media objects", Information modelling and knowledge bases, 579-90,1990.
- [ 7 ] Y. Tanaka, "A Toolkit System for the Synthesis and Management of Active Media Objects", Deductive and Object-Oriented Databases, Proc. of the 1st Int'l conf. , 259-77, 1990.
- [ 8 ] Y. Tanaka, A. Nagasaki, M. Akaishi and T. Noguchi, "A Synthetic Media Architecture for an Object-Oriented Open Platform", IFIP Trans. A, Comput. Sci. Technol., 104-10,1992.
- [ 9 ] 中川光紀,"シンセティックメディアによるフォームフローシステムの構築", 北海道大学大学院工学研究科修士課程電気工学専攻, 応用制御工学講座, 平成6年度修士論文.
- [10] UniSQL/X™データベース管理システムユーザーズマニュアル, NTTデータ通信株式会社.

[12] Objectworks\Smalltalk User's Guide, ParcPlace Systems.

## 付録 1. UniSQLで管理するデータの構造定義

### 1. NRDF

#### 1. 1 用語の定義と説明

データストリーム：NRDFデータベースの1件分のデータの記述全体を指す。一件の論文にして発表された一連の実験から得られた1つ、または複数の数値データテーブルとその内容を理解すために必要な書誌的内容・実験条件・測定物理の記述をその内容とする。

セクション：一つのデータストリームのなかで、記述の重複が無いように、書誌的内容、実験条件、数値データテーブルとそのパラメータ情報に内容を分けて、記述ブロックをつくる。この記述ブロックをセクションという。

データセット：一つのデータテーブルとその物理的および書誌的内容を理解するために必要な情報の集合はデータセットといわれる。一つのデータセットは、一つのデータテーブルを含む一つのDATAセクションおよび一つまたは複数のEXPとBIBセクションの集合である。

#### 1. 2 UniSQLデータベースオブジェクトの定義

##### 1. 2. 1 作成するクラスの概要

NRDFデータに対して、データストリーム、セクション、およびデータセットの各レベルでそれぞれクラスを定義する。

###### (1) NrdfDB クラス

このクラスでは、格納しているデータストリームを定義する。

###### (2) NrdfSection クラス

このクラスは、NrdfDBのサブクラスとして定義され、セクション本体のデータを保持する。NRDFの各セクションはNrdfSectionクラスの一つのインスタンスとして格納する。一つのデータストリームの各セクションに対して、データベースへの格納に際してその順序番号を割り当てる。この順序番号は、sectionNoとして参照される。

### (3) NrdfValueIndexクラス

NrdfValueIndexクラスは、データセットを記述している属性と値をそのインスタンスとして、この各インスタンスがどのデータセットに属するかを定義する。このクラスはデータセットに対する索引を与える。インスタンスは、各値につき作成する。

#### 1. 2. 2 クラス定義の詳細

##### (1) NrdfDB クラス

クラス名：NrdfDB

属性定義：

属性名	属性定義
dId	char(1)
dIdNo	smallint

各属性の内容

NRDFのデータストリームには、これを識別する番号が割り当てられている。この番号は先頭の英字1文字と数字の一連番号の部分から構成されている。この番号はまた、このデータストリーム格納しているファイル名に使われている。

dId	ファイル名の先頭の一字で、Dである。
dIdNo	データストリームに割り当てられた一連番号である。

##### (2) NrdfSectionクラス

クラス名：NrdfSection as subclass of NrdfDB

属性定義：

属性名	属性定義
sectionKin	char(6)
datasetList	set(smallint)

sectionNo	smallint
sectionBody	sequence(char(80))

各属性の内容

sectionKind	このセクションの種類を表す. ￥￥BIB   ￥￥EXP   ￥￥DATA
sectionNo	このセクションに割り当てられた順序番号
sectionBody	セクション本体の記述

(3) NrdfValueIndexクラス

クラス名: NrdfValueIndex as subclass of NrdfDB

属性の定義:

属性名	属性定義
datasetList	set(integer)
attribSQ	smallint
aFlag	set(char(2))
attrib	char(20)
valSQ	smallint
evalSQ	smallint
vFlag	set(char(2))
tValue	char(20))
fText	sequence(char(80))

各属性の内容

datasetList	value または ftext の値が含まれるデータセットの集合である.
attribSQ	値が複合文で定義された場合は, 定義している文の複合文内の出現順序を表す. 単文の場合は0とする.
aFlag	文に対するフラグである.

attrib	文の左辺の値, または, 次の値とする. COMMENT ftextにコメントを入れた場合 DATAHEADING valがデータヘッディングの場合 DATAUNIT valがデータ単位の場合
valSQ	valまたはftextが値のセット与える文のものであったとき, その値が定義されたセット内の順序. 単一の値の時は0とする. データヘディングまたはデータユニットの場合はテーブルにおけるカラム位置を表す.
evalSQ	値が拡張記号の場合, 拡張記号内における位置を示す. 先頭は0, 括弧内は順に1から, 括弧の外は, -1とする.
vFlag	val または ftext に割り当てられているフラグ
tValue	値, 拡張記号の場合は, その個々の要素である.
fText	フリーテキストの場合の値.

## 2. EXFOR

### 2. 1 用語の定義

- work : 一連の実験からなる一つもまとまった仕事をさす。
- subwork : workを構成する個々の実験に相当し、一つの数値テーブルを生成する。
- ENTRY : EXFORで記述されたひとつのwork
- SUBENTRY : EXFORで記述されたsubworkである。ただし、先頭のSUBENTRYは特別で、そのENTRYを構成する他のすべてのSUBENTRYで共通の情報が抜き出されて記述される。

### 2. 2 UniSQLデータベースオブジェクトの定義

#### 2. 2. 1 作成するクラスの概要

EXFORデータに対して、ENTRYとSUBENTRYのレベルでそれぞれクラスを定義する。

##### (1) ExforDBクラス

このクラスは、格納しているENTRYを定義する。

##### (2) ExforSectionクラス

このクラスはExforDBのサブクラスとして定義され、EXFORの記述データ自身を保持する。そのインスタンスは、SUBENTRYを単位とし、各インスタンスはこれを構成するSECTIONの記述自身の記述からできている。

##### (3) ExforValueIndexクラス

属性とその値の対をそのインスタンスとして持ち、それがどのsubworkに属するかを定義する。このクラスはsubworkに対する索引テーブルを与える。

#### 2. 2. 2 クラス定義の詳細

##### (1) ExforDBクラス

クラス名 : ExforDB

属性の定義：

属性名	属性定義
OriginId	char(1)
EntryNo	smallint

各属性の内容

OriginId	ENTRYに割り当てられているデータ識別子の先頭の1文字
EntryNo	ENTRYに割り当てられているデータ識別子の先頭の1文字を除く順序番号

## (2) ExforSectionクラス

クラス名：ExforSection as subclass of ExforDB

属性の定義：

属性名	属性定義
subEntryNo	smallint
bibSec	sqquence(ExfLine)
commonSec	sequence(ExfLine)
dataSec	sequence(ExfLine)

各属性の内容

subEntryNo	SUBENTRY番号
bibSec	BIBセクションの記述
commonSec	COMMONセクションの記述
dataSec	DATAセクションの記述

## (3) ExforValueIndexクラス

クラス名：ExforValueIndex as subclass of ExforDB

属性の定義：

属性名	属性定義
subEntryNo	smallint
attribSQ	smallint
aFlag	set(char(2))
attrib	char(20)
valSQ	smallint
eValSQ	smallint
vFlag	set(char(2))
tValue	char(20)
fText	sequence(char(80))

#### 各属性の内容

subEntryNo	SUBENTRY番号
attribSQ	属性の記述が構造化されている場合、値が属性の構造のどの位置を占めるかその位置を示す
aFlag	属性の構造の型
attrib	属性キーワード
valSQ	属性に複数のコードまたはフリーテキストが割り当てられている場合、そのコードまたはフリーテキストの順番
eValSQ	コードの記述が構造化されている場合、切り出したコードの構造上の位置
vFlag	コードに付加されたフラグ
tValue	コード
fText	フリーテキスト



TYPE	F V S C E W
CLASS	1 2 3 4 5 6 7 8 9 10 11 12 13 14 2桁までの数字，一桁の数字の場合，先頭は空白，複数ある時は「,」で区切ってならべる。
RATE	実数 E12.5
BASE	CLASS=14の場合に書かれるCODE
SOURCE	EXFOR NRDF
DATE	YY-MM-DDの形式で書かれる日付
COUNTRY	4桁の国別コード
FLAG	使わなくなったコードを0で示す

### 3. 2 NRDF辞書の属性

NRDF辞書定義に要するすべての属性とその定義をあげる。

属性名	属性の定義
CODE	CHAR(20)
EXPANSION	VCHAR(72)
EXPLANATION	SEQUENCE CHAR(72)
TYPE	CHAR(1)
CLASS	SET SMALLINT
RATE	REALまたは CHAR
BASE	CHAR(20)
SOURCE	CHAR(6)
UPDATE	DATE または CHAR(8)
COUNTRY	CHAR(4)
FLAG	CHAR(2)

### 3. 3 NRDF辞書クラス

NRDFの各辞書はこれらの属性の部分集合で表すことができる。これら辞書の

データを管理するため、共通する属性はNrdfDictionクラスで定義し、それぞれのタイプの辞書はこのサブクラスとして定義する。

(1) NrdfDiction

code	char(40) not null
expansion	char(80)
explanation	sequence(char(80))
dtype	char(1) not null
udate	char(8)
oflag	char(1)

(注) udate の内容は、dd-mm-yy の形式の文字列である。

(2) NrdfDictionC as subclass of NrdfDiction

dtype	char(1) shared 'C'
-------	--------------------

(3) NrdfDictionE as subclass of NrdfDiction

dtype	char(1) shared 'E'
-------	--------------------

(4) NrdfDictionF as subclass of NrdfDiction

dtype	char(1) shared 'F'
-------	--------------------

(5) NrdfDictionS as subclass of NrdfDiction

dtype	char(1) shared 'S'
-------	--------------------

(6) NrdfDictionV as subclass of NrdfDiction

dtype	shared 'V'
codeclass	set(small int)
rate	char(12)

base char(40)

source char(5)

country char(7)

(注) RATE の内容は, {, }9.99999E{+, -}99 の形式の文字列である.

(7) NrdfDictionW as subclass of NrdfDiction

dtype shared 'W'

#### 4. EXFOR辞書

##### 4. 1 辞書のファイルの形式と内容

EXFOR辞書のファイルはEXFORデータと同様の形式でつくられている。それぞれのキーワードまたはコードの種類ごとに一つの辞書に編成されていて、各辞書には1から50までの番号が割り当てられている。各辞書は基本的に次の情報項目を持っている：(1) 定義を与えるキーワードまたはコードと(2) その説明、(3) および定義用語が今では使われなくなったこと等示すフラグである。そして、(2)の説明は(a)プレーテキスト、(b)キーワードまたはコードの展開形、(c)展開形とフリーテキストのいずれかであたえられている。しかし、各辞書によって、これら情報の与え型が異なる場合、およびこれらの情報に付加する情報項目がある場合がある。

それで、各辞書に共通する情報項目は、ExforDictionクラスで定義し、各辞書はそれぞれその番号を名前の一部にもつサブクラスとして定義する。

##### 4. 2 EXFOR辞書のクラスと各クラスの属性

###### 4. 2. 1 ExforDictionクラスの属性定義

DICNO	SMALLINT
FLAG	CHAR(1)
CODE	CHAR(10)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))

###### 4. 2. 2 各個別辞書の辞書番号、内容と情報を定義する属性

EXFORの各辞書のクラスはExforDictionのサブクラスとして定義する。

D01 : System Identifiers

FLAG	CHAR(1)
CODE	CHAR(10)

EXPLANATION SEQUENCE(CHAR(55))

\*D02 : Information Identifiers

FLAG	CHAR(1)
CODE	CHAR(10)
EXPANSION	CHAR(22)
EXPLANATION	SEQUENCE(CHAR(55))
KWREQ	CHAR(11)
CODEINF	CHAR(11)
DICTPOINT	CHAR(11)

D03 : Institute Codes

FLAG	CHAR(1)
CODE	CHAR(7)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))

D04 : Reference Types

FLAG	CHAR(1)
CODE	CHAR(1)
EXPANSION	CHAR(4)
EXPLANATION	CHAR(44)
DICTPOINT	CHAR(6)

\*D05 : Journal Codes

FLAG	CHAR(1)
CODE	CHAR(6)
EXPANSION	CHAR(50)

EXPLANATION	SEQUENCE(CHAR(55))
COUNTRY	CHAR(4)

D06 : Reports

FLAG	CHAR(1)
CODE	CHAR(11)
EXPLANATION	SEQUENCE(CHAR(55))
INSTITUTE	CHAR(7)

\*D07 : Bokks, Conference Codes

FLAG	CHAR(1)
CODE	CHAR(10)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))
LEXPANSION	SEQUENCE(CHAR(55))

D08 : Elements

FLAG	CHAR(1)
CODE	CHAR(6)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))

\*D09 : Compounds

FLAG	CHAR(1)
CODE	CHAR(10)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))

\*D13 : Particles

FLAG	CHAR(1)
CODE	CHAR(3)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))

D15 : History Codes

FLAG	CHAR(1)
CODE	CHAR(1)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))

D16 : Status Codes

FLAG	CHAR(1)
CODE	CHAR(5)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))
REQACCF	CHAR(1)
COMMENT	SEQUENCE(CHAR(55))

D17 : Related Reference Code

FLAG	CHAR(1)
CODE	CHAR(1)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))

D18 : Facility Codes

FLAG	CHAR(1)
------	---------

CODE	CHAR(5)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))
CATEGORY	CHAR(55)

D19 : Incident Source Codes

FLAG	CHAR(1)
CODE	CHAR(5)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))
CATEGORY	CHAR(55)

\*D20 : Additional Result Codes

FLAG	CHAR(1)
CODE	CHAR(5)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))

D21 : Method Codes

FLAG	CHAR(1)
CODE	CHAR(5)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))
CATEGORY	CHAR(55)

D22 : Detector Codes

FLAG	CHAR(1)
CODE	CHAR(5)

EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))
CATEGORY	CHAR(55)

D23 : Analysis Codes

FLAG	CHAR(1)
CODE	CHAR(5)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))
CATEGORY	CHAR(55)

\*D24 : Data Headings

FLAG	CHAR(1)
CODE	CHAR(10)
EXPLANATION	SEQUENCE(CHAR(55))
CATEGORY	CHAR(55)
CHECKFLAG	CHAR(1)

D25 : Data Units

FLAG	CHAR(1)
CODE	CHAR(10)
EXPLANATION	SEQUENCE(CHAR(55))
CATEGORY	CHAR(55)
DIMENSION	CHAR(4)
CONVFACT	CHAR(11)

D27 : Nuclides

FLAG	CHAR(1)
------	---------

CODE	CHAR(10)
STATE	CHAR(13)
EXPLANATION	SEQUENCE(CHAR(55))

D28 : Inc. Particles (REACTION SF2)

FLAG	CHAR(1)
CODE	CHAR(3)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))

D29 : Prod. Particles (REACTION SF3)

FLAG	CHAR(1)
CODE	CHAR(3)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))

D30 : Processes (REACTION SF 3)

FLAG	CHAR(1)
CODE	CHAR(3)
EXPLANATION	SEQUENCE(CHAR(55))

D31 : Branch Codes (REACTION SF 5)

FLAG	CHAR(1)
CODE	CHAR(5)
EXPLANATION	SEQUENCE(CHAR(55))

D32 : Parameters (REACTION SF 6)

FLAG	CHAR(1)
------	---------

CODE	CHAR(3)
EXPLANATION	SEQUENCE(CHAR(55))
CATEGORY	CHAR(55)

D33 : Particle Consid. (REACTION SF7)

FLAG	CHAR(1)
CODE	CHAR(3)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))

D34 : Modifiers (REACTION SF 8)

FLAG	CHAR(1)
CODE	CHAR(3)
EXPLANATION	SEQUENCE(CHAR(55))
COMMENT	SEQUENCE(CHAR(55))

D35 : Data Types (REACTION SF 9)

FLAG	CHAR(1)
CODE	CHAR(5)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))

\*D36 : Quantities (REACTION SF 5-8)

FLAG	CHAR(1)
CODE	CHAR(44)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))
CATEGORY	CHAR(55)

DIMENSION	CHAR(3)
RESONANCEP	CHAR(1)
LXPANSION	SEQUENCE(CHAR(55))

\*D37 : Result Codes

FLAG	CHAR(1)
CODE	CHAR(5)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))

D42 : CINDA Quantities

FLAG	CHAR(1)
CODE	CHAR(3)
EXPANSION	CHAR(55)
EXPLANATION	SEQUENCE(CHAR(55))
NCODE	SMALLINT

D43 : NLIB for evaluated libralies

FLAG	CHAR(1)
LIBNO	SMALLINT
LIBNAME	CHAR(55)

D50 : List of Dictionaries

FLAG	CHAR(1)
CODE	SMALLINT /* DICTION# */
EXPANSION	CHAR(33) /* DICTNAME */
EXPLANATION	SEQUENCE(CHAR(55))
	/* COMMENT FOR THIS DICT */

UPDATE	CHAR(6)	/* YYMMDD */
CODELENGTH	SMALLINT	
CODEEXPANF	CHAR(3)	